LA-UR-91- 1840

LA-UR--91-1840

DE91 013386

TITLE: PROGRESS IN THE DEVELOPMENT OF A NUMERICAL TOKAMAK

AUTHOR(S): Jeremiah U. Brackbill, T-3

SUBMITTED TO *Engineering Research / Power Supercomputer Users Symposium*
Gaithersburg, MD
May 21-22, 1991

Los Alamos National Laboratory
Los Alamos, New Mexico 87545

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Progress in the Development of a Numerical Tokamak

J. U. Brackbill
Los Alamos National Laboratory
Los Alamos, NM 87545

I. The Numerical Tokamak

The development of highly concurrent computers is causing a revolution in numerical modeling. One can now anticipate a time when constraints on computational power are removed. That anticipation motivates the identification of grand challenge problems, those problems which combine significance and generality, and benefit from enormous computing power.

Magnetic confinement fusion faces a grand challenge problem that will stimulate the energy and creativity of the entire community of computational physicists. That challenge is the development of a numerical model for a Tokamak. The successful modeling of a Tokamak would lead to a better understanding of the operation of these machines. As John Dawson comments, "it would give much greater confidence in the predicted machine performance; it would provide the capability to test out our new ideas and machine improvements without committing them to hardware, and it would be a powerful tool for machine builders." Since success with the Tokamak is the principal hope for magnetic fusion in the forseeable future, real contributions from numerical modeling would be invaluable.

The challenge in modeling Tokamaks lies in the variety of physical processes that occur, and the complexity of their interaction. There is at the core of a Tokamak a hot, collisionless plasma confined by magnetic fields, and sustained by microwave and neutral beam heating. There is at the wall of a Tokamak a cold, collisionless plasma where atomic physics and surface physics determine the impurity levels and overall energy confinement. Both of these regions of the Tokamak present problems whose solution is a major challenge. The complexity of a calculation that treats both regions simultaneously must be addressed from the outset. Software tools must be developed that allow scientists with many interests to work together, so that, for example, one can ask how a neutral beam injector modifies the radial plasma profile, and thus th magnetohydrodynamic stability; how large excursions from the reactor operating point affect the lifetime of the first wall, or how to match the properties of the magnet power supply to the inductance of the plasma.

There is agreement on certain essential components of Tokamak physics. The modeling of Tokamaks requires the means to represent complex geometries, long-range interaction of charged particles through electric and magnetic fields, non-equilibrium plasma phenomena, and the nonlinear interaction of many complex and dissimilar processes. It is clear that no one method can encompass all of the physics of a Tokamak. However, because of the range of plasma conditions encountered in a tokamak, there is reason to believe that plasma simulation using the full kinetic equations has the greatest chance of achieving this goal [1].

Plasma simulation uses particle methods to model the dynamics of a plasma in a magnetic field. From a computational point of view, particle methods are challenging candidates for computing on highly concurrent computers. Their successful implementation requires tackling problems in load balancing and communication whose solution will drive the development of new software and hardware tools and point the way for a wide variety of computational algorithms and applications. In addition, particle methods have found very wide application, including turbulence modeling, tracer transport in global climate modeling, biological flows, and space plasmas, all of which will benefit from the development of a numerical Tokamak.

II. Plasma Simulation On Magnetohydrodynamic Time Scales

To understand the kinetic processes that determine energy confinement and impurity production in magnetic confinement experiments, one must solve the plasma kinetic equations on the time scales of magnetic fusion experiments. This is a major challenge, both in size of the computational problem and its complexity. One not only must increase the capabilities of plasma simulation by developing methods that can be used on the highly parallel computers of the future, but also must incorporate the best of the methods that are currently available. For example, the implicit method that is described below extended the range of plasma simulation a hundredfold. The capabilirv to model kinetic effects in three dimensions on magnetohydrodynamic time scales in realistic geometries is the ultimate goal.

In a numerical calculation, the equations of motion for a plasma in a static magnetic field include an equation to calculate the particle position, $\mathbf{x}_p$, from the particle velocity, $\mathbf{u}_p$

$$\mathbf{x}_p^1 = \mathbf{x}_p^0 + \mathbf{u}_p^{1/2}\Delta t , \qquad \mathbf{x}_p^0 = \mathbf{x}_p(t),\ \mathbf{x}_p^1 = \mathbf{x}_p(t+\Delta t),\ \mathbf{u}_p^{1/2} = \frac{1}{2}\left[\mathbf{u}_p(t) + \mathbf{u}_p(t+\Delta t)\right], \tag{1}$$

a momentum equation to calculate the particle velocity in the electric, $\mathbf{E}$, and magnetic, $\mathbf{B}$, fields,

$$\mathbf{u}_p^{1/2} = \mathbf{u}_p^0 + \frac{q_s}{m_s}\left[\mathbf{E}^\theta + \frac{\mathbf{u}_p^{1/2}\times\mathbf{B}}{c}\right]\frac{\Delta t}{2} , \tag{2}$$

(where q_s/m_s is the charge to mass ratio, and c is the speed of light), and Poisson's equation to calculate the long-range interaction among the particles through an electric field,

$$\nabla \bullet \mathbf{E}^\theta = 4\pi n^\theta . \tag{3}$$

where Δt is the time step and n^θ and $\mathbf{E}^\theta$ are the charge density and electric field at $(t+\theta\Delta t)$. The particles are labeled by the subscript p, and the species (e.g. ions and electrons) by the subscript s. The charge density is calculated by interpolating the particle charge to the grid with a particle shape, S.

$$n^\theta = \sum_s \sum_p \int d^3x'\, S(\mathbf{x}-\mathbf{x}';h)\, q_s\, \delta(\mathbf{x}'-\mathbf{x}_p^\theta)/h^3 . \tag{4}$$

$\mathbf{x}_p^\theta$, for example, is given by, $\mathbf{x}_p=(1-\theta)\,\mathbf{x}_p(t)+\theta\,\mathbf{x}_p(t+\Delta t)$. Evaluating (4) from the particle data, solving (3), then solving (1) and (2) advances the solution from t to t+ Δt.

When $\theta=0$, the difference equations (1-4) are explicit and the time step is limited by a linear stability condition which depends on the plasma frequency, ω_p. When $\omega_p\Delta t<2$, the difference equations are stable. When $\omega_p\Delta t>2$, the equations are unstable. Since ω_p usually is much higher than the frequency of interest, one pays a very high price to model low-frequency phenomena because the time step is so small.

Implicit difference equations eliminate the stability constraint. When $1/2<\theta<1$, Eqs. (1-4) are implicit and the equations are stable even with very large Δt. However, one cannot solve Eqs. (1-4) with $\theta>1/2$. Instead, one must derive fluid-like equations by expanding the charge density about $\mathbf{x}_p^0$ in powers of $\delta=k<\delta\mathbf{x}^2>^{1/2}$, where k is a characteristic wave number, and $<\delta\mathbf{x}^2>$ is the mean square particle displacement in a time step. The expansion introduces higher order moments of the particles, including the current density, $\mathbf{J}$, and the pressure, Π, but the sequence of moments can be truncated by neglecting terms of $O(\delta^2)$. From the moment equations one can evaluate n^θ in terms of known quantities and derive an equation for the electric field to replace Eq (3),

$$\nabla \bullet \varepsilon \bullet \mathbf{E} = 4\pi n' \tag{5}$$

where n' is calculated from the particle data. The dielectric tensor, ε, has the form,

$$\varepsilon \bullet \mathbf{E}^\theta = (1 + \varepsilon_{perp})\,\mathbf{E}^\theta + \varepsilon_{trans}\,\mathbf{E}^\theta \times \mathbf{B} + \varepsilon_{par}\,\mathbf{E}^\theta \bullet \mathbf{B}\,\mathbf{B} . \tag{6}$$

Various terms are defined in Vu and Brackbill[3], and implicit plasma simulation is reviewed in Brackbill and Cohen[2]. All of the intermediate variables are easily calculated.

The dielectric tensor contains information to anticipate the response of the plasma to the electric field, so that while the right side contains information at time t, the electric field can be calculated at time $t+\Delta t$. When the field equation, Eq. (5), is replaced by a finite difference approximation to model an inhomogeneous and magnetized plasma, one must invert a non-symmetric matrix with variable coefficients. We use an incomplete Cholesky decomposition and conjugate gradient iteration[4] to solve the field equation in CELESTE.The best solver for a highly concurrent computer is a research question.

III. Plasma Simulation in Real Geometries

To do plasma simulation in real geometries requires many changes in the basic method[5]. In CELESTE, a code developed at Los Alamos, the particle equations of motion are solved on a structured grid of arbitrarily shaped, six-sided cells. For example, a body-fitted grid tmay be used o model the geometry of magnetic confinement experiments by causing the computational domain to conform to the structure and shape of the boundary as shown in Fig. 1, where a mesh for a torus is shown[6].

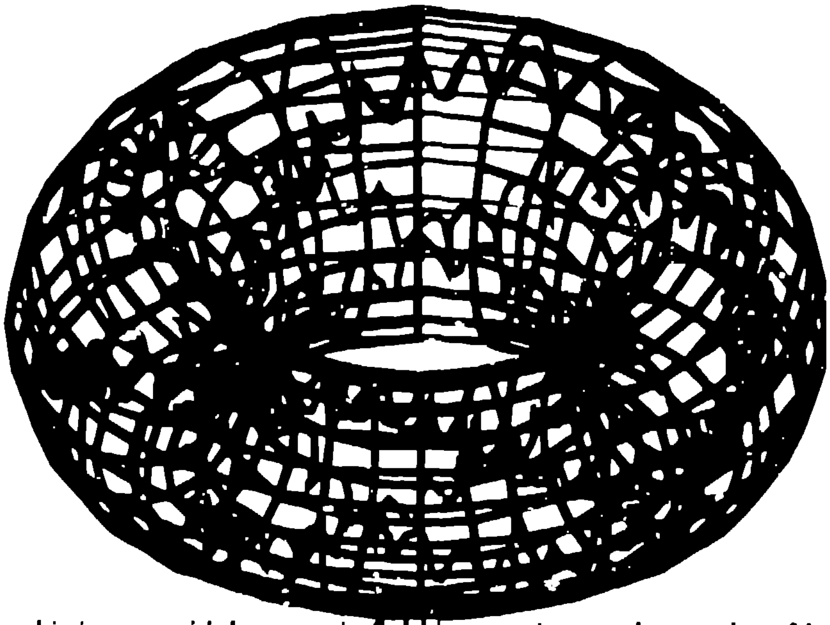

Fig. 1 A particle orbit in a toroidal magnetic field is superimposed on a plot of he outer surface of the computation grid on which th orbit is calculated.

The solution of the plasma simulation equations on an arbitrary grid uses both physical and natural coordinates. The natural coordinates (ξ,η,ν), give the location on the grid, and are calculated by mapping each of CELESTE's six-sided cells in physical space on to a unit cube in natural space. At each vertex, the natural coordinates assume integer values, (i,j,k), which are constant as the mesh moves. Elsewhere, the mapping between physical and natural coordinates is given by trilinear interpolation.

The grid may be completely redefined from one cycle to the next, except that the new grid also must map on to a logical cube. In CELESTE, the number of grid points is fixed, but the grid points may be moved about as needed. This allows one to use an adaptive grid.

Adaptive grids have been applied extensively in modeling singular phenomena in two dimensions. A partial list of applications includes magnetic reconnection, shocks, and hydrodynamically unstable interfaces. The use of adaptive grids has resulted in more accurate numerical solutions than would have been possible with uniform grids.The adaptive grid is generated by solving a variational problem, in which one minimizes a functional[7]. A simple functional is,

$$I_s = \sum_{i=1}^{3} \iiint dx\, dy\, dz\, w \; (\nabla \xi^i)^2 \,, \quad (\xi^1, \xi^2, \xi^3) = (\xi, \eta, \nu) \qquad (7)$$

With $w=1$, minimizing I_s gives a body-fitted grid like the one shown in Fig. 1. When w is not constant, the grid points are closer together where w is larger, and further apart where w is smaller. If w is an appropriate function of the data, the grid will adapt to provide increased resolution where it is needed by moving grid points. For example, if w is proportional to the current density in a calculation of flow in a time-dependent magnetic field, the grid will cluster to resolve steep gradients in the magnetic field intensity [8].

Body-fitted grids for fluid and magnetohydrodynamic flows have been used for many years[9]. Their use for particle simulations is a new development[10, 11], and new methods for moving particles through the grid have been developed. In a recent method developed by Mike Jones at Los Alamos, the particle orbit is calculated on the body-fitted grid using an algorithm that can be implemented easily on a concurrent processor. The orbit is calculated in natural coordinates, which are generated by mapping each cell in the mesh to a unit cube. The natural coordinates locate the particle within the mesh, and eliminate the expensive local calculations that are otherwise necessary on a non-rectilinear grid[12, 13].The physical coordinates of a particle are advanced by solving Eq. (1). The natural coordinates of a particle are calculated using the covariant and contravariant mesh vectors [6].

The covariant and contravariant vectors are defined by,

$$\mathbf{a}_i = \frac{\partial \mathbf{x}}{\partial \xi^i}, \tag{8}$$

$$\mathbf{a}^i = \nabla \xi^i = \frac{\mathbf{a}_j \times \mathbf{a}_k}{J} \tag{9}$$

where (i, j, k) = (1, 2, 3) in cyclic permutation and the Jacobian, J, is defined by,

$$J = (\det(g_{ij}))^{\frac{1}{2}} = \mathbf{a}_i \bullet (\mathbf{a}_j \times \mathbf{a}_k) , \tag{10}$$

The natural coordinates of a particle are advanced from the equation,

$$\frac{d\xi^i}{dt} = \frac{d\mathbf{x}}{dt} \bullet \mathbf{a}^i \tag{11}$$

The covariant mesh vectors are computed on the grid and interpolated to the particles as needed.

IV. Some Early Results

A computational study of edge effects in magnetized plasmas has yielded the discovery of high frequency sheath oscillations with implications for future experiments[14].

In CELEST1D, a one-dimensional code[3], the full electromagnetic interactions are solved directly with arbitrarily large time steps. With small time steps, the accuracy of an implicit calculation is as good as an explicit calculation using the standard leapfrog differencing in time. With large time steps, the accuracy and stability of the implicit method are much better than the explicit method.

In two dimensions, transport driven by the ion temperature gradient instability is calculated with CELESTE [15]. Similarly to Lee, a 16x16 cell mesh with 64 particles per cell was used[16]. The plasma is magnetized with a strong field perpendicular to the plane of the calculation, and a small component in the vertical, or periodic direction. Temperature and density gradients are maintained in the horizontal direction. The time step is $\omega_{pe}\Delta t=2500$, 1250 times the explicit stability limit, and $\Omega_{ci}\Delta t=1.36$. The calculation is continued for 800 time steps, or 173 ion gyroperiods. Linear theory predicts instability when the ratio, η_i, of the ion temperature to the ion density gradient length scales is greater than one. In Fig. 2, a selected ion orbit is plotted with $\eta_i=0.5$, which is stable. Each point corresponds to the position of the ion on a time step. The ion drifts along the magnetic field direction, with little motion in the horizontal direction. (When the ion leaves the top boundary, it returns through the bottom.)

When η_i is increased to 2, the same particle executes the orbit shown in Fig. 3. Because of the electrostatic field caused by the ion-temperature gradient instability, there is now lateral motion by the ions. This motion leads to increased heat transfer from the hot boundary on the right to the cold boundary on the left. The heat transfer is between 5 and 10 times that calculated from classical theory.

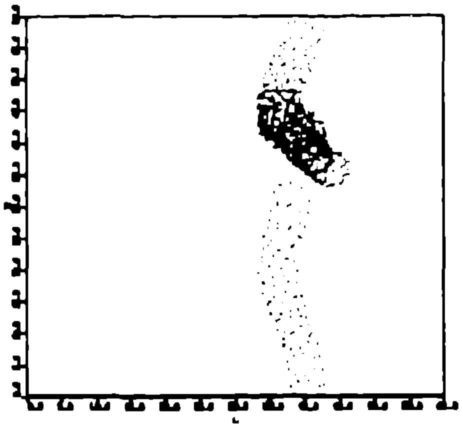

Figure 2. An ion orbit for an stable plasma is plotted.

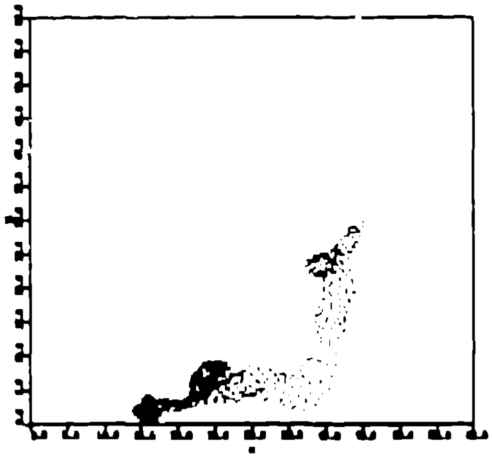

Figure 3. An ion orbit in an unstable plasma is plotted.

In Fig. 1, a particle orbit in a static magnetic field with winding number 1 is shown. The orbit is calculated on the toroidal grid whose outer surface is shown. The temperature is high to make the gyromotion visible.

References

1. Brackbill, J.U. & Forslund, D.W. in *Multiple Time Scales* 271-310 (Academic Press, Orlando, 1985).
2. Brackbill, J.U. & Cohen, B.I. *Multiple Time Scales* 1-442 (Academic Press, Orlando, 1985).
3. Vu, H.X. & Brackbill, J.U. *Computer Physics Communications* (1991-submitted).
4. Jordan, T.J. in *Parallel Computations* 1 (Academic Press, Orlando, 1982).
5. Brackbill, J.U. & Ruppel, H.M. *J. Comput. Phys.* 314 (1986).
6. Thompson, J.F., Warsi, Z.U.A. & Mastin, C.W. *Numerical Grid Generation: Foundations and Applications* (Elsevier, New York 1985).
7. Brackbill, J.U. & Saltzman, J.S. *J. Comput. Phys.* **46**, 342-367 (1982).
8. Milroy, R.D. & Brackbill, J.U. *Phys. Fluids* **25**, 775 (1982).
9. Brackbill, J.U. (eds. Killeen, J.) 1 (Academic Press, New York, 1976).
10. Brackbill, J.U., Kothe, D.B. & Ruppel, H.M. *Comput. Phys. Comm.* **48**, 25-38 (1988).
11. Westermann, T. **A263**, 271-279 (1988).
12. Lohner, R. & Ambrosiano, J. *J. Comput. Phys.* **91**, 22-31 (1990).
13. Westermann, T. *J. Comput. Phys-submitted* (1989).
14. Myra, J.R. *et al.*, *Phys. Rev. Lett.* Mar. 4 (1991).
15. Horton, W.J., Estes, R.D. & Biskamp, D. *Plasma Phys.* **22**, 663 (1980).
16. Lee, G.S. & Diamond, P.H. *Physics of Fluids* **29**, 3291 (1986).